database-travel

April 12, 2025

[1]:

```python
import pandas as pd
from sqlalchemy import create_engine
```

[2]:

```python
import numpy as np
```

[3]:

```python
# Load dataset from a CSV file
rating_df = pd.read_csv('data 2.csv')
distance_df=pd.read_csv('data 5.csv')
place_df=pd.read_csv('new datak.csv')
```

[ ]:

```python

```

[21]:

```python
# Create a database connection using SQLAlchemy
engine = create_engine('mysql+pymysql://root:SiKu17@26R@localhost/
 ↪TravelRecommendationDB2')
```

[ ]:

```python

```

[24]:

```python
import pymysql

# Use pymysql to establish the connection
mydb = pymysql.connect(
    host='localhost',       # Replace with '127.0.0.1' if localhost fails
    user='root',            # Your MySQL username
    password='SiKu17@26R',  # Your MySQL password
    port=3306               # MySQL port, typically 3306
)

print("Connected successfully!")
```

```
Connected successfully!
```

[ ]:

```python

```

[25]:

```python
import pymysql

try:
    connection = pymysql.connect(
        host='localhost',
        user='root',
        password='SiKu17@26R',  # Use original password here
        database='TravelRecommendationDB2'
    )
    print("Connected to MySQL successfully!")
except pymysql.MySQLError as e:
    print("Error: ", e)
```

```
Connected to MySQL successfully!
```

[ ]:

```
```

[26]:

```
from sqlalchemy import create_engine, text

# Connect to your MySQL database
engine = create_engine("mysql+pymysql://root:SiKu17%4026R@localhost:3306/
 ↪TravelRecommendationDB2")

# SQL to create Places table
create_places_table = """
CREATE TABLE IF NOT EXISTS Places (
    Place_Id INT PRIMARY KEY,
    Place_Name VARCHAR(255),
    Age VARCHAR(20),
    Category VARCHAR(100),
    Road_condition VARCHAR(20),
    Weather_Condition VARCHAR(50),
    Description TEXT,
    Mode_of_Transport VARCHAR(255),
    Latitude FLOAT,
    Longitude FLOAT
);
"""

# SQL to create Ratings table
create_ratings_table = """
CREATE TABLE IF NOT EXISTS Ratings (
    User_Id INT,
    Place_Id INT,
    Rating FLOAT,
    PRIMARY KEY (User_Id, Place_Id),
    FOREIGN KEY (Place_Id) REFERENCES Places(Place_Id)
);
"""
```

```
# SQL to create Distances table
create_distances_table = """
CREATE TABLE IF NOT EXISTS Distances (
    Place_Id INT PRIMARY KEY,
    Source VARCHAR(100),
    Destination VARCHAR(100),
    Distance_km FLOAT,
    FOREIGN KEY (Place_Id) REFERENCES Places(Place_Id)
);
"""

# Execute the commands
with engine.connect() as connection:
    connection.execute(text(create_places_table))
    connection.execute(text(create_ratings_table))
    connection.execute(text(create_distances_table))

print("  All tables created successfully.")
```

```
 All tables created successfully.
```

[ ]:

```

```

[9]:

```
#from sqlalchemy import create_engine, text

# Create connection
#engine = create_engine("mysql+pymysql://root:SiKu17%4026R@localhost:3306/
 ↪TravelRecommendationDB")

# Drop tables in the correct order (due to foreign key constraints)
#drop_ratings_table = "DROP TABLE IF EXISTS Ratings;"
#drop_distances_table = "DROP TABLE IF EXISTS Distances;"
#drop_places_table = "DROP TABLE IF EXISTS Places;"

# Execute the drop commands
#with engine.connect() as connection:
    #connection.execute(text(drop_ratings_table))
    #connection.execute(text(drop_distances_table))
    #connection.execute(text(drop_places_table))

#print("  All tables dropped successfully.")
```

```
 All tables dropped successfully.
```

[ ]:

```

```

[30]:
```
with engine.connect() as connection:
    connection.execute(text("ALTER TABLE Places MODIFY Age VARCHAR(50);"))    ␣
 ↪  #modify this table
```

[32]:
```
distance_df.rename(columns={"Distance(km)": "Distance_km"}, inplace=True)    ␣
 ↪ #modify this table
```

[48]:
```
rating_df.rename(columns={'Place_rating': 'Rating'}, inplace=True)  #  modify␣
 ↪this table
```

[35]:
```
# Make sure your column names in the DataFrames match the SQL table columns

# Save Places data
place_df.to_sql(name='Places', con=engine, if_exists='append', index=False)

# Save Ratings data
#rating_df.to_sql(name='Ratings', con=engine, if_exists='append', index=False)

# Save Distances data
#distance_df.to_sql(name='Distances', con=engine, if_exists='append',␣
 ↪index=False)

print("  Data inserted successfully into all tables!")
```

```
 Data inserted successfully into all tables!
```

```
C:\Users\shaw3\AppData\Local\Temp\ipykernel_4760\2085570752.py:4: UserWarning:
The provided table name 'Places' is not found exactly as such in the database
after writing the table, possibly due to case sensitivity issues. Consider using
lower case table names.
  place_df.to_sql(name='Places', con=engine, if_exists='append', index=False)
```

[36]:
```
import mysql.connector
from mysql.connector import errorcode
```

[37]:
```
from mysql.connector import Error
```

[39]:
```
try:

    # Load data into SQL tables
    place_df.to_sql('Places', con=engine, if_exists='append', index=False)
    #distances_df.to_sql('Distances', con=engine, if_exists='append',␣
 ↪index=False)
    #ratings_df.to_sql('Ratings', con=engine, if_exists='append', index=False)

    print("Data successfully loaded into SQL tables!")
except Exception as e:
    print("Error loading data into SQL tables:", e)
```

```
Error loading data into SQL tables: (pymysql.err.IntegrityError) (1062,
"Duplicate entry '1' for key 'places.PRIMARY'")
[SQL: INSERT INTO `Places` (`Place_Id`, `Place_name`, `Age`, `Category`,
`Road_condition`, `Weather_Condition`, `Description`, `Mode_of_Transport`,
`Latitude`, `Longitude`) VALUES (%(Place_Id)s, %(Place_name)s, %(Age)s,
%(Category)s, %(Road_condition)s, %(Weather_Condition)s, %(Description)s,
%(Mode_of_Transport)s, %(Latitude)s, %(Longitude)s)]
[parameters: [{'Place_Id': 1, 'Place_name': 'Victoria Memorial', 'Age': 'All
Ages', 'Category': 'Historical Monument', 'Road_condition': 'Good',
'Weather_Condition': 'Haze', 'Description': 'A grand white marble monument
dedicated to Queen Victoria, featuring Indo-Saracenic architecture and housing a
museum with artifacts from British India.', 'Mode_of_Transport': 'Bus, Taxi',
'Latitude': 22.54498, 'Longitude': 88.34243}, {'Place_Id': 2, 'Place_name':
'Quest Mall', 'Age': 'All Ages', 'Category': 'Shopping, Entertainment',
'Road_condition': 'Good', 'Weather_Condition': 'Haze', 'Description': 'A modern,
upscale shopping mall with various brands, restaurants, and entertainment
options.', 'Mode_of_Transport': 'Bus, Taxi, Metro', 'Latitude': 22.53915,
'Longitude': 88.36603}, {'Place_Id': 3, 'Place_name': 'Fort William Kolkata',
'Age': 'All Ages', 'Category': 'Historical Site', 'Road_condition': 'Good',
'Weather_Condition': 'Cloudy', 'Description': 'A historic British fort with a
museum showcasing military history and a serene park for picnics.',
'Mode_of_Transport': 'Bus, Taxi', 'Latitude': 22.55895, 'Longitude': 88.33773},
{'Place_Id': 4, 'Place_name': 'Shalimar Station', 'Age': 'All Ages', 'Category':
'Transportation Hub (Railway Station)', 'Road_condition': 'Good',
'Weather_Condition': 'Hazr', 'Description': 'A major railway station in Howrah,
Kolkata, serving local and long-distance trains.', 'Mode_of_Transport': 'Train,
Bus, taxi', 'Latitude': 22.55591, 'Longitude': 88.31503}, {'Place_Id': 5,
'Place_name': 'Belur Math', 'Age': 'All Ages', 'Category':
'Religious/Spiritual', 'Road_condition': 'Good', 'Weather_Condition': 'Haze',
'Description': 'The headquarters of the Ramakrishna Mission, a spiritual
organization founded by Swami Vivekananda.', 'Mode_of_Transport': 'Bus, Taxi,
Ferry', 'Latitude': 22.63282, 'Longitude': 88.35642}, {'Place_Id': 6,
'Place_name': 'Howrah Bridge', 'Age': 'All Ages', 'Category': 'Architectural
Landmark', 'Road_condition': 'Good', 'Weather_Condition': 'Cloudy',
'Description': 'A majestic suspension bridge connecting Kolkata to Howrah,
offering stunning views of the Hooghly River.', 'Mode_of_Transport': 'Bus, Taxi,
Walking', 'Latitude': 22.58532, 'Longitude': 88.34681}, {'Place_Id': 7,
'Place_name': 'Birla Planetarium', 'Age': 'Children, Teens, Families',
'Category': 'Science/Education', 'Road_condition': 'Good', 'Weather_Condition':
'Clear', 'Description': 'A popular science museum with interactive exhibits and
shows on astronomy and space science.', 'Mode_of_Transport': 'Bus, Taxi',
'Latitude': 22.54548, 'Longitude': 88.34732}, {'Place_Id': 8, 'Place_name':
'Indian Museum', 'Age': 'Children, Teens, Families', 'Category': 'Museum',
'Road_condition': 'Average', 'Weather_Condition': 'Clear', 'Description': 'One
of the oldest museums in India, housing a vast collection of artifacts from
various cultures and periods.', 'Mode_of_Transport': 'Bus, Taxi, Metro',
'Latitude': 22.55108, 'Longitude': 88.35109}  … displaying 10 of 170 total
bound parameter sets …  {'Place_Id': 169, 'Place_name': 'Atmosphere', 'Age':
'All Ages', 'Category': 'floating bridge', 'Road_condition': 'Average',
'Weather_Condition': 'Haze', 'Description': "It is a sky bridge, called Deya, is
the world's first residential floating\xa0sculpture", 'Mode_of_Transport': 'Bus,
Taxi , Auto-rickshaw', 'Latitude': 22.63792, 'Longitude': 88.45364},
{'Place_Id': 170, 'Place_name': "Abanindranath Tagore's Garden House", 'Age':
'All Ages', 'Category': 'Heritage House, Garden, Art, Museum (Possible)',
'Road_condition': 'Average', 'Weather_Condition': 'Haze', 'Description': 'The
former residence and garden of the renowned artist Abanindranath Tagore,
possibly preserved as a museum or heritage site.', 'Mode_of_Transport': 'Bus,
Taxi , Auto-rickshaw', 'Latitude': 22.7051, 'Longitude': 88.3445}]]
(Background on this error at: https://sqlalche.me/e/20/gkpj)
```

[40]:

```python
pd.read_sql("SELECT * FROM Places LIMIT 5", con=engine)
```

[40]:

```
   Place_Id             Place_Name       Age  \
0         1      Victoria Memorial  All Ages
1         2             Quest Mall  All Ages
2         3  Fort William Kolkata  All Ages
3         4       Shalimar Station  All Ages
4         5             Belur Math  All Ages

                              Category Road_condition Weather_Condition  \
0                  Historical Monument           Good              Haze
1              Shopping, Entertainment           Good              Haze
2                      Historical Site           Good            Cloudy
3  Transportation Hub (Railway Station)           Good              Hazr
4                  Religious/Spiritual           Good              Haze

                                         Description Mode_of_Transport  \
0  A grand white marble monument dedicated to Que…         Bus, Taxi
1  A modern, upscale shopping mall with various b…  Bus, Taxi, Metro
2  A historic British fort with a museum showcasi…         Bus, Taxi
3  A major railway station in Howrah, Kolkata, se…  Train, Bus, taxi
4  The headquarters of the Ramakrishna Mission, a…  Bus, Taxi, Ferry

   Latitude  Longitude
0   22.5450    88.3424
1   22.5392    88.3660
2   22.5590    88.3377
3   22.5559    88.3150
4   22.6328    88.3564
```

RATING DATA LOAD SUCCESSFULLY

[49]:

```
# Save Ratings data
rating_df.to_sql(name='Ratings', con=engine, if_exists='append', index=False)

# Save Distances data
#distance_df.to_sql(name='Distances', con=engine, if_exists='append',␣
 ↪index=False)

print("  Data inserted successfully into all tables!")
```

```
  Data inserted successfully into all tables!
```

```
C:\Users\shaw3\AppData\Local\Temp\ipykernel_4760\2836253830.py:2: UserWarning:
The provided table name 'Ratings' is not found exactly as such in the database
after writing the table, possibly due to case sensitivity issues. Consider using
lower case table names.
  rating_df.to_sql(name='Ratings', con=engine, if_exists='append', index=False)
```

[42]:

```
rating_df.head()
```

[42]:

```
   User_Id  Place_Id  Place_rating
0        5         1           4.1
1       40         2           4.2
2    11799         3           4.6
3       81         4           3.1
4       69         5           3.7
```

[ ]:

```

```

DISTANCES TABLE LOAD DATA

[45]:

```
distance_df.to_sql(name='Distances', con=engine, if_exists='append',␣
 ↪index=False)

print("  Data inserted successfully into all tables!")
```

```
  Data inserted successfully into all tables!
```

```
C:\Users\shaw3\AppData\Local\Temp\ipykernel_4760\2383541724.py:1: UserWarning:
The provided table name 'Distances' is not found exactly as such in the database
after writing the table, possibly due to case sensitivity issues. Consider using
lower case table names.
  distance_df.to_sql(name='Distances', con=engine, if_exists='append',
index=False)
```

[ ]:

```

```

[ ]:

```

```